

## CLAIMS

1. A method for statement boundary detection comprising:

obtaining an input stream;

5 parsing said input stream to determine a natural end of a first statement using a programming language syntax; and  
dividing said input stream into a series of statements wherein said natural end is used to divide said first statement from a second statement.

10 2. The method of claim 1 wherein said parsing comprises:

retrieving a next character from said input stream; and

positioning said natural end of said first statement immediately before said next character, if appending said next character to said first statement is inconsistent with said programming language syntax.

15 3. The method of claim 1 wherein said parsing comprises:

determining a context from one or more characters previously retrieved from said input stream; and

positioning said natural end based on said context and said programming language  
20 syntax.

4. The method of claim 1 further comprising:

detecting a statement termination token.

25 5. A statement boundary detector comprising:

an obtainer configured to obtain an input stream;

a parser configured to parse said input stream to determine a natural end of a first statement using a programming language syntax; and

a partitioning unit configured to divide said input stream into a series of

5 statements wherein said natural end is used to divide said first statement from a second statement.

6. The statement boundary detector of claim 5 wherein said parser comprises:

a retrieval unit configured to retrieve a next character from said input stream; and

10 a positioning unit configured to position said natural end of said first statement immediately before said next character, if appending said next character to said first statement is inconsistent with said programming language syntax.

7. The statement boundary detector of claim 5 wherein said parser comprises:

15 a determiner configured to determine a context from one or more characters previously retrieved from said input stream; and

a positioning unit configured to position said natural end based on said context and said programming language syntax.

20 8. The statement boundary detector of claim 5 further comprising:

a detector configured to detect a statement termination token.

9. A computer program product comprising:

a computer usable medium having computer readable program code embodied therein configured to detect a statement boundary, said computer program product comprising:

computer readable code configured to cause a computer to obtain an input stream;

5 computer readable code configured to cause a computer to parse said input stream to determine a natural end of a first statement using a programming language syntax; and

computer readable code configured to cause a computer to divide said input stream into a series of statements wherein said natural end is used to divide said first statement from a second statement.

10

10. The computer program product of claim 9 wherein said computer readable code configured to cause a computer to parse comprises:

computer readable code configured to cause a computer to retrieve a next character from said input stream; and

15

computer readable code configured to cause a computer to position said natural end of said first statement immediately before said next character, if appending said next character to said first statement is inconsistent with said programming language syntax.

11. The computer program product of claim 9 wherein said computer readable code configured to cause a computer to parse comprises:

20

computer readable code configured to cause a computer to determine a context from one or more characters previously retrieved from said input stream; and

computer readable code configured to cause a computer to position said natural end based on said context and said programming language syntax.

25

12. The computer program product of claim 9 further comprising:  
computer readable code configured to cause a computer to detect a statement  
termination token.

LA 48583v3